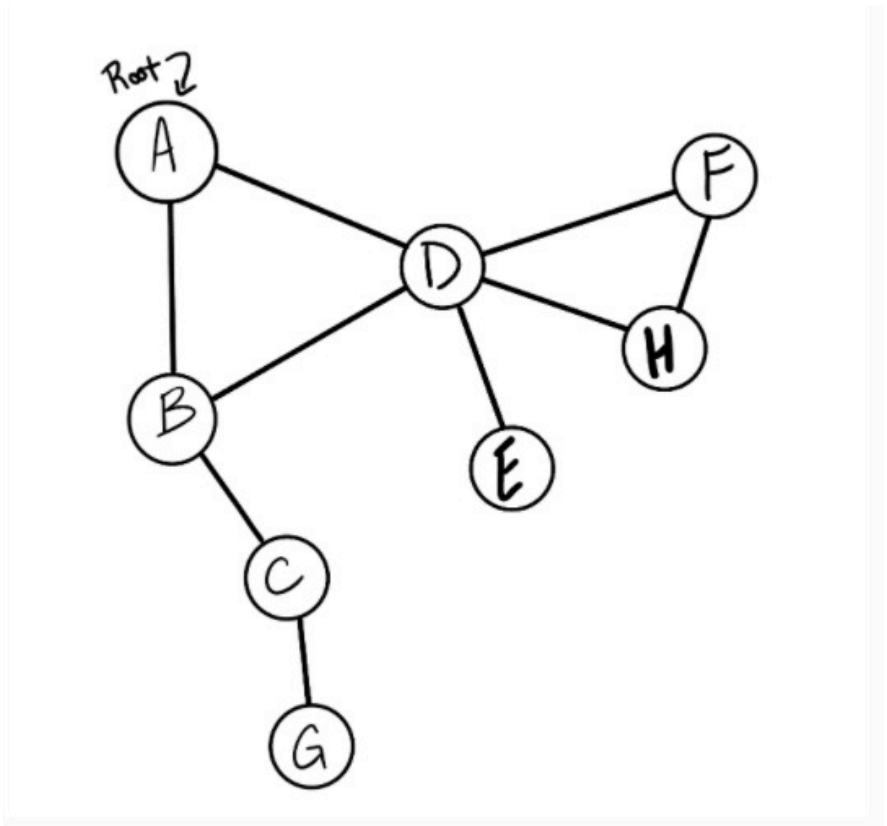
Problem domain.

Write a function that accept an adjancency list as a graph, and return a collection of nodes in thier preorder depth first traversal order.

Illustration.



Output: A, B, C, G, D, E, H, F

Algorith.

- Writing a function that take a list as a parameter.
- Create a variable call arr.
- creat a function that take vertex
- check if that vertex is in arr.

- return if it's the case.
- if not, push that vertex in arr.
- loop through all the neighbor of the vertex and do the same.

Pseudo-code / code.

```
Def    depth_first ( vertex)
  List_ vertex = []
      walk(root)
        if vertex in List_vertex
               return.
          List_vertex.append(vertex)
          Nebor = [ edge. vertex for edge in getNeighbor ( vertex) ]
            for vertex in Nebor.
                walk (vertex)
        walk (vertex)
        return List_vertex.
```